Desktop Connection Library

Yes, your shiny new computer can talk to your old Newton

A bit of history

- Newton Connection Utilities
- The DILs (Desktop Integration Library legacy)
- Mac OS “Classic” and the Newton
- What do we lack to have a native OSX connection utility?

A bit of history
NCU

- Apple in-house development
- Can do :
  - Backup / Restore
  - Keyboard
  - Import / Export data
  - Install packages
- Can’t do :
  - Software integration
  - Adapt to new data type

A bit of history
DILs

- Apple development library for connectivity (Desktop Integration Library)
- Limited to network connections (serial and OpenTransport)
- Led to Thomas Tempelmann's "DIL Tester", a first form of NCU replacement

A bit of history
Mac OS “Classic” and the Newton

- Why can’t we simply use NCU under Classic?
- Why can’t we simply use the DILs under Classic for a NCU revival?
- What is lacking in Mac OS X to just make the whole thing work?

A bit of history
What do we need for a NCU revival?

- Connectivity : the Newton “native” connection scheme works only on AppleTalk or serial
- Data : we need to have a way to go back and forth with data on the Newton
- Long term objective : cross platform Newton connection kit

The DCL is a communication tool

- Status of the communication layers available in the DCL
- Status of the communication protocols available in the DCL

DCL : communication layers

	Serial	IrDA	Bluetooth	AppleTalk	TCP	Zeroconf
Mac OS	√	×	×	√	√	×
Mac OS X	√	×	√	√	√	√
UNIX	√	×	√	×	√	×
Windows	×	×	×	×	×	×

DCL : Available protocols

Browse	√
Install	√
Backup/ Restore	×
Keyboard	√
Synchronize	×
File Import/ Export	×

The DCL is a “Newton framework”

- Understanding Newton objects : a bit of NewtonScript
- Newton objects in the DCL : NSOF
- Import / Export examples

Understanding Newton objects

	Properties	Examples
Immediates (numeric values)	base blocks booleans, integers, floating point,...	42 0x2A 101010
Arrays	ordered collections of objects	[1,2,3,4,5]
Strings	array of characters (UTF16 on the Newton)	“Hello World”
Frames (dictionaries)	“set” of key/value pairs	{ name:“Nicolas Zinovieff”, alias:“Krugazor” }

DCL : NSOF

	Properties	Examples
Immediates (numeric values)	Objects as well	TDCLNSRef::MakeInt(42);
Arrays	Same semantics	newArray->Add(TDCLNSRef::MakeInt(42));
Strings	Same semantics, but conversion needed	TDCLNSRef::MakeString("Hello World")
Frames (dictionaries)	Same semantics	newFrame->Set(TDCLNSRef::MakeSymbol("alias"), TDCLNSRef::MakeString("Krugazor"));

DCL : Newton Objects Examples
NWT → XML

```
// Create in/out files from the platform dependant classes

// Open input file
inInputFile.Open( true /* inReadOnly */ );

// Open output file
((TDCLFile*) theOutputFile)->Open( false /* inReadOnly */ );

// create decoder
TDCLNSOFDecoder theDecoder( &inInputFile );

// create encoder
TDCLXMLEncoder theEncoder( theOutputFile );

// convert
theEncoder.AddObject(theDecoder.GetNextObject());

// close files
inInputFile.Close();
theOutputFile->Close();
```

Next step : improvements

- We have connectivity and data interpretation : what's next?
- Synchronization
- Trojans
- Future work

Implementation examples

- NCU - like (Escale, Delivery, ...)
- Data driven (PBBookMaker, Notes converter, ...)
- Demo